

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-152551

(43)Date of publication of application : 16.06.1995

(51)Int.Cl.

G06F 9/06

(21)Application number : 05-299630

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 30.11.1993

(72)Inventor : MIURA MASAHIRO

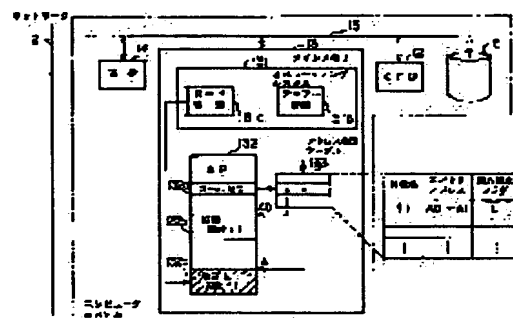
(54) COMPUTER SYSTEM AND PROGRAM EXECUTING METHOD

(57)Abstract:

PURPOSE: To realize the on-line version up by correcting a function without discontinuing the execution of a program.

CONSTITUTION: An address control table 133 contains the entry addresses for each function, and an entry address is selected out of the table 133 when the corresponding function is carried out. Meanwhile a corrected function is loaded in a memory 13 and an entry address of the table 133 is rewritten into the entry address of the loaded function. Thus a program can be corrected without discontinuing the program.

Therefore it is just required for revision of a certain function to add a new version of the function with its old version left in a program as it is and to rewrite the address corresponding to the function into the address of new version of the function. In such a way, the on-line version up can easily be attained.



LEGAL STATUS

[Date of request for examination] 07.09.1998

[Date of sending the examiner's decision of rejection] 20.11.2001

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平7-152551

(43)公開日 平成7年(1995)6月16日

(51)Int.Cl.⁶

G 0 6 F 9/06

識別記号

5 4 0 E 9367-5B
F 9367-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数4 O L (全 7 頁)

(21)出願番号 特願平5-299630

(22)出願日 平成5年(1993)11月30日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 三浦 雅弘

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

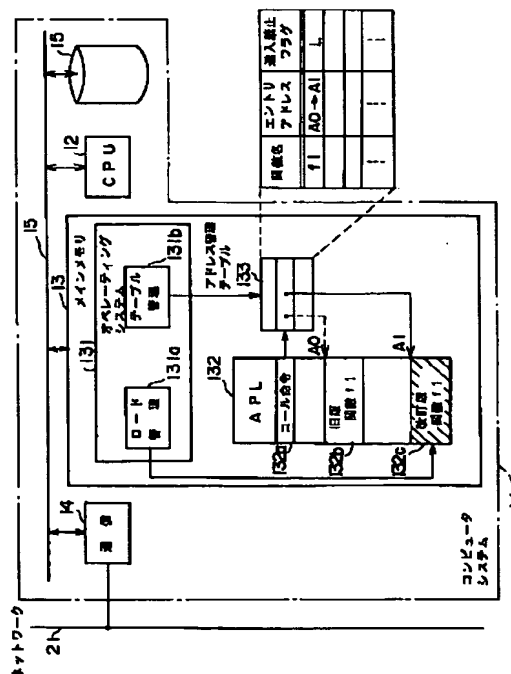
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 コンピュータシステムおよびプログラム実行方法

(57)【要約】

【目的】プログラム実行を停止せずに関数を修正できるようにし、オンライン・バージョンアップの実現を図る。

【構成】関数毎にエントリアドレスを保持するアドレス管理テーブル133が設けられており、関数を実行する際にはアドレス管理テーブル133から該当するエントリアドレスが求められ、また関数を修正する場合には修正した関数がメモリ13上にロードされ、該アドレス管理テーブル133のエントリアドレスをロードした関数のエントリアドレスに書き換えることによって、プログラムを停止することなく、プログラムの修正が行われる。したがって、ある関数を改版する際には、該関数の旧版はプログラム中に残したまま新版をプログラムに追加し、該関数に対応するアドレスを、新版関数のものに書き換えるだけでよい。このことにより、オンライン・バージョンアップが容易に実現できる。



【特許請求の範囲】

【請求項 1】 実行対象のプログラムに含まれる各サブブルーチン毎にそのサブブルーチンが格納されているメモリ上のエントリアドレスを示すアドレス情報を保持するアドレス管理テーブルと、

このアドレス管理テーブルを参照して呼び出し対象のサブブルーチンに対応するメモリ上のエントリアドレスを検出し、そのエントリアドレスのサブブルーチンを呼び出して実行するサブブルーチン呼び出し手段と、

修正対象のサブブルーチンに対応する新たなサブブルーチンをメモリ上にロードし、前記アドレス管理テーブルの該当するエントリアドレスを、前記新たなサブブルーチンをロードしたエントリアドレスに書き替えるテーブル更新手段とを具備することを特徴とするコンピュータシステム。

【請求項 2】 前記アドレス管理テーブルは、前記各サブブルーチン毎にそのサブブルーチンの呼び出し禁止を示す呼び出し禁止情報がセットされる領域を有し、前記テーブル更新手段は、前記アドレス管理テーブルのエントリアドレスの書き替え期間中、前記サブブルーチン呼び出し手段による前記修正対象サブブルーチンの呼び出しが禁止されるように、前記エントリアドレスの書き替えに先立って前記アドレス管理テーブルに前記呼び出し禁止情報をセットする手段を有することを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 3】 実行対象のプログラムに含まれる各サブブルーチン毎にそのサブブルーチンが格納されているメモリ上のエントリアドレスを示すアドレス情報と、そのサブブルーチンの呼び出し禁止および許可を示す 2 つのステートを持つ呼び出し制御情報とを保持するアドレス管理テーブルと、

前記各サブブルーチン毎にそのサブブルーチンを実行しているスレッド数を示すスレッドカウンタと、

前記アドレス管理テーブルを参照して呼び出し対象のサブブルーチンに対応するメモリ上のエントリアドレスを検出し、そのエントリアドレスのサブブルーチンを呼び出して実行させるサブブルーチン呼び出し手段と、

修正対象のサブブルーチンに対応する新たなサブブルーチンをメモリ上にロードし、前記アドレス管理テーブルの該当するエントリアドレスを、前記新たなサブブルーチンをロードしたエントリアドレスに書き替えるテーブル更新手段と、

前記サブブルーチン呼び出し手段による前記修正対象サブブルーチンの呼び出しが禁止されるように、前記エントリアドレスの書き替えに先立って前記アドレス管理テーブルに呼び出し禁止ステートの呼び出し制御情報をセットする手段と、

前記スレッドカウンタを参照して前記修正対象サブブルーチンを実行しているスレッド数が無くなるのを検出し、その検出に回答して前記呼び出し制御情報を呼び出し許

可ステートに書き替える手段とを具備することを特徴とするコンピュータシステム。

【請求項 4】 実行対象のプログラムに含まれる各サブブルーチン毎にそのサブブルーチンが格納されているメモリ上のエントリアドレスを示すアドレス情報をメモリ上に登録し、

前記実行対象のプログラムのサブブルーチン呼び出し命令に回答して、前記アドレス情報を参照して呼び出し対象のサブブルーチンに対応するメモリ上のエントリアドレスを検出し、

その検出されたエントリアドレスのサブブルーチンを呼び出して実行し、

修正対象のサブブルーチンに対応する新たなサブブルーチンをメモリ上にロードし、

前記アドレス情報の該当するエントリアドレスを、前記新たなサブブルーチンをロードしたエントリアドレスに書き替えることを特徴とするプログラム実行方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明はコンピュータシステムおよびプログラム実行方法に関し、特にそのシステムにおけるプログラムの訂正・修正を行うのに適したコンピュータシステムおよびプログラム実行方法に関する。

【0002】

【従来の技術】近年、コンピュータの性能は飛躍的に向上し続けており、それにつれてプログラムの大規模化・高速化が進んでいる。このため、プログラムのバグも不可避免的に増加している。

【0003】コンピュータとそのプログラムは企業情報システムにおいて重要な役割を果たしている場合が多く、そのようなプログラムのバグは、発見次第ただちに訂正することが重要である。

【0004】また、プログラムへの要求の変化などといった要因により、プログラムの機能を変更／改良したい場合も多い。従来のプログラム実行方法では、サブブルーチンを呼出す際は、リンク時に決定される該サブブルーチンのアドレスを直接用いて呼出している。この方法では、プログラムを変更すると一般にサブブルーチンのアドレスも変わるため、サブブルーチンを呼出す箇所（サブブルーチンのエントリアドレス）もそれぞれ変更する必要がある。このため、動作中のプログラムの改版（オンライン・バージョンアップと称する）を行うのは非常に困難であり、プログラムを一旦停止して、コンパイルし直すといった作業が必要とされる。このことは、多大な損失をもたらすことがあり、特に銀行などのオンラインシステムで使用するコンピュータにおいて口座の出入金管理などのを行うプログラムを改訂する場合には、オンラインシステム全体の稼働を停止しなければならないという不具合が発生する。

【0005】

【発明が解決しようとする課題】従来では、プログラムを変更するとサブルーチンのエントリアドレスも変わるためオンライン・バージョンアップが困難であり、プログラムを修正する場合にはシステムを一旦稼働停止して、プログラムをコンパイルし直さなければならないという欠点があった。

【0006】この発明はこのような点に鑑みてなされたもので、システムを稼働停止すること無く容易にオンライン・バージョンアップを行うことができるコンピュータシステムおよびプログラム実行方法を提供することを目的とする。

【0007】

【課題を解決するための手段および作用】この発明によるコンピュータシステムは、実行対象のプログラムに含まれる各サブルーチン毎にそのサブルーチンが格納されているメモリ上のエントリアドレスを示すアドレス情報を保持するアドレス管理テーブルと、このアドレス管理テーブルを参照して呼び出し対象のサブルーチンに対応するメモリ上のエントリアドレスを検出し、そのエントリアドレスのサブルーチンを読み出して実行させるサブルーチン呼び出し手段と、修正対象のサブルーチンに対応する新たなサブルーチンをメモリ上にロードし、前記アドレス管理テーブルの該当するエントリアドレスを、前記新たなサブルーチンをロードしたエントリアドレスに書き替えるテーブル更新手段とを具備することを特徴とする。

【0008】このコンピュータシステムにおいては、サブルーチン毎にエントリアドレスを保持するアドレス管理テーブルが設けられており、サブルーチンを実行する際にはアドレス管理テーブルから該当するエントリアドレスが求められ、またサブルーチンを修正する場合には修正したサブルーチンをメモリ上にロードし、該アドレス管理テーブルのエントリアドレスをロードしたサブルーチンのエントリアドレスに書き換えるだけによって、プログラムを停止することなく、プログラムの修正が行われる。

【0009】したがって、あるサブルーチンを改版する際には、該サブルーチンの旧版はプログラム中に残したまま新版をプログラムに追加し、該サブルーチンに対応するアドレスを、新版サブルーチンのものに書き換えるだけでよい。このことにより、オンライン・バージョンアップが容易に実現できる。

【0010】また、アドレス管理テーブルには、サブルーチンの呼び出し禁止を示す呼び出し禁止情報がセットされる領域が設けられており、エントリアドレスの書き換え期間中は、サブルーチン呼び出し手段による旧版のサブルーチンの呼び出しが禁止されるように、呼び出し禁止情報がセットされる。これにより、スレッド間のプログラム実行内容の整合性を保証することができる。

【0011】

【実施例】以下、図面を参照してこの発明の実施例を説明する。図1にはこの発明の一実施例に係わるコンピュータシステムの構成が示されている。このコンピュータシステム11は、オンラインシステムを構築する複数のコンピュータシステムの1つであり、ネットワーク21を介して他のシステムと接続されている。

【0012】コンピュータシステム11には、図示のように、CPU12、メインメモリ13、通信制御ユニット14、および磁気ディスク装置15が設けられており、これらはシステムバス15を介して相互接続されている。

【0013】CPU12は、このコンピュータシステム11全体を制御するためのものであり、メインメモリ13のオペレーティングシステム131や各種アプリケーションプログラムを実行する。メインメモリ13は、前述のオペレーティングシステム131、および実行対象のアプリケーションプログラム132を格納する。

【0014】オペレーティングシステム131は、システム管理のための各種プログラムから構成されており、ここには、この発明のプログラム実行方法を実現するために必要なサブルーチンロード管理プログラム131aとテーブル管理プログラム131bが組み込まれている。

【0015】サブルーチンロード管理プログラム131aは、改訂された新たなサブルーチンを磁気ディスク装置15またはネットワーク21からメインメモリ13の空き領域にローディングする。テーブル管理プログラム131bは、アドレス管理テーブル133をメモリ13上に確保し、そのテーブル133の内容を管理する。このテーブル管理プログラム131bは、サブルーチンロード管理プログラム131aによって新たなサブルーチンがメインメモリ13にローディングされると、アドレス管理テーブル133の旧版のサブルーチンに対応するエントリアドレスをロードアドレスに書き替える。

【0016】アプリケーションプログラム132は、口座の出入金管理を行うためのものであり、複数のサブルーチンを含んでいる。このアプリケーションプログラム132は言語Cによって記述されている。言語Cでは、サブルーチンは関数と称されている。したがって、以下の記載ではサブルーチンをすべて関数と称することにする。

【0017】アプリケーションプログラム132においては、関数コール命令132aは関数(f1)を呼び出す命令であるが、その呼び出し先関数のエントリアドレスは関数コール命令132aには組み込まれておらず、そのエントリアドレスはアドレス管理テーブル133を参照することによって求められる。アドレス管理テーブル133が関数(f1)のエントリアドレスとして“A0”を示す場合には、関数(f1)132bが呼び出されて実行されるが、関数(f1)のエントリアドレスと

して“ A 1 ”を示す場合には、関数 (f 1) 1 3 2 c が呼び出されて実行される。

【 0 0 1 8 】アドレス管理テーブル 1 3 3 は、アプリケーションプログラム 1 3 2 に含まれる各関数毎に、その識別子 (関数名)、メモリ 1 3 のエントリアドレス、および進入禁止フラグを保持する。関数 (f 1) を修正する場合には、改訂版関数 (f 1) 1 3 2 c がローディングされた後、アドレス管理テーブル 1 3 3 の関数名 f 1 に対応するエントリアドレスが図示のように A 0 から A 1 に書き換えられる。進入禁止フラグは、関数修正期間中は“ H ”にセットされ、これによって関数修正時に旧版の関数が呼び出されることが防止される。

【 0 0 1 9 】通信制御ユニット 1 4 は、ネットワーク 2 1 を介して他のシステムとの通信を実行するためのものであり、例えば、改訂版関数をネットワーク 2 1 からダウンロードするために用いられる。磁気ディスク装置 1 5 は、コンピュータシステム 1 1 の 2 次記憶装置として利用されるものであり、各種プログラムやファイルが格納されている。

【 0 0 2 0 】次に、図 2 を参照して、アドレス管理テーブル 1 3 3 の具体的な内容の一例を説明する。図示のように、アプリケーションプログラム 1 3 2 の関数 read file () 、 check data () 、 write file () を有する場合、アドレス管理テーブル 1 3 3 には、それら関数名とエントリアドレスとの対応が保持される。ここでは、関数 read file () 、関数 check data () 、関数 write file () が、それぞれアドレス 0 x 7 0 0 0 0 0 0 0 、 0 x 7 0 0 0 1 0 0 0 、 0 x 7 0 0 0 5 0 0 0 に存在する場合が想定されている。

【 0 0 2 1 】関数 read file () を改訂する場合には、その関数 read file () の改訂版がメモリ上にロードされ、関数 read file に対応するエントリアドレスが旧版関数 read file () に対応する 0 x 7 0 0 0 0 0 0 0 から改訂版関数 read file のエントリアドレス (ここでは、 0 x 7 0 0 0 6 0 0 0) に変更される。以降、関数 read file () を呼び出す際には、アドレス 0 x 7 0 0 0 6 0 0 0 がコールされる。

【 0 0 2 2 】図 3 には、従来の C プログラムの関数呼び出し命令と本発明で採用され関数呼び出し命令とが比較して示されている。通常の C プログラムにおいては、関数 read file () をコールする場合には、引数 file をもって関数 read file () を呼出すコード片がコール命令として使用される。この場合、関数 read file () のアドレス 0 x 7 0 0 0 0 0 0 0 は、リンク時にオブジェクトコード中に埋め込まれる。

【 0 0 2 3 】これに対し、この実施例では、関数名の文字列を引数として使用し、アドレス管理テーブル 1 3 3 を参照して該関数のアドレスを返す関数 fn name to adrs () を介して関数 read file () を呼出すコード片、すなわち、 fn name to adrs (" read file") (file)

が、関数 read file () のコール命令として使用される。

【 0 0 2 4 】次に、図 4 のフローチャートを参照して、関数 read file () を改版するための動作手順を説明する。まず、オペレーティングシステム 1 3 1 のテーブル管理プログラム 1 3 1 b は、旧版の関数 read file () が新たに実行されないようにするために、関数 read file () に対応する進入禁止フラグを“ H ”にセットし、この関数の呼出しを一時的に禁止する (ステップ S 1 1) 。次いで、ロード管理プログラム 1 3 1 a は、関数 read file () の新版のオブジェクトコードをメモリ 1 3 の空き領域 (アドレス 0 x 7 0 0 0 6 0 0 0 以降) にローディングして、新版の関数 read file () を追加する (ステップ S 1 2) 。

【 0 0 2 5 】この後、テーブル管理プログラム 1 3 1 b は、アドレス管理テーブル 1 3 3 の関数 read file () に対応するエントリアドレスを、新版のアドレス 0 x 7 0 0 0 6 0 0 0 に書換え (ステップ S 1 3) 、そして、フラグを“ H ”から“ L ”にリセットすることによってその関数の呼出しを再び許可する (ステップ S 1 4) 。

【 0 0 2 6 】この結果、旧版の関数 read file () のオブジェクトコードを変更しなくても、新版の read file () を呼出すことが可能となる。なお、この図 4 の手順は、関数 read file () の改版中における新たな呼出しだけを一時停止しているもので、例えばマルチスレッドやマルチタスクといったマルチプログラミングにおいては、改版後に旧版と新版の read file () が同時に動作するといった状況も発生し得る。この状況は必ずしも不都合を招くものではないが、図 5 のフローチャートに示されているように、図 4 のフローチャートにステップ S 1 0 0 を追加することにより、このような状況の発生を容易に防止することができる。

【 0 0 2 7 】すなわち、図 5 においては、図 4 のフローチャートと同様にステップ S 1 1 ~ S 1 3 が実行されるが、ステップ S 1 0 0 によって関数 read file () を実行しているすべてスレッドの処理が終了するのが待たれ、その後に新版の関数の呼び出し許可が許可される (ステップ S 1 4) 。

【 0 0 2 8 】このステップ 1 0 0 は、各関数毎にその関数を実行しているスレッド (またはタスク) の数をカウントするスレッドカウンタを用意することによって実現できる。このスレッドカウンタは、オペレーティングシステム 1 3 1 の機能の一部として実現することができる。この場合、アドレス管理テーブル 1 3 3 は、図 6 に示されているように、スレッドカウンタの値を保持するフィールドを持つことが好ましい。

【 0 0 2 9 】次に、図 7 および図 8 のフローチャートを参照して、関数 read file () を呼出す手順を説明する。まず、呼出すべき関数 read file () のアドレスを求めるために、関数名の文字列“ read file ”を引数と

して、関数 fn name to addr() が呼出される（ステップ S 2 1）。

【0030】このステップ S 2 1 では、図 8 に示されているように、まず、引数として与えられた関数名 “read file” に対応する関数 read file() が現在改版中であるかどうかを調べるために、進入禁止フラグを検査される（ステップ S 2 1 1）。該フラグが “H” にセットされていれば改版中であるので、リセットされるまでループして待つことにより、該関数の呼出しが一時的に停止される。そして、関数名 “read file” に対応するアドレスがアドレス管理テーブル 1 3 3 をサートすることによって求められ、その求めたアドレスが呼び側へ返される（ステップ S 2 1 2, S 2 1 3）。

【0031】この後、図 7 に示されているように、fn name to addr() から帰ってきた read file() のアドレスを用いて、関数 read file() が呼出される（ステップ S 2 2）。

【0032】以上のように、この実施例においては、関数毎にエントリアドレスを保持するアドレス管理テーブル 1 3 3 が設けられており、関数を実行する際にはアドレス管理テーブル 1 3 3 から該当するエントリアドレスが求められ、また関数を修正する場合には修正した関数がメモリ上にロードし、該アドレス管理テーブル 1 3 3 のエントリアドレスをロードした関数のエントリアドレスに書き換えることによって、プログラムを停止することなく、プログラムの修正が行われる。

【0033】したがって、ある関数を改版する際には、該関数の旧版はプログラム中に残したまま新版をプログラムに追加し、該関数に対応するアドレスを、新版関数のものに書換えるだけでよい。このことにより、オンライン・バージョンアップが容易に実現できる。

【0034】また、アドレス管理テーブル 1 3 3 には、関数の呼び出し禁止を示す呼び出し禁止フラグがセットされる領域が設けられており、エントリアドレスの書き換え期間中は、旧版関数の呼び出しが禁止されるように、呼び出し禁止フラグがセットされる。これにより、スロット間のプログラム実行内容の整合性を保証することができる。

【0035】

【発明の効果】以上のように、この発明によれば、あるサブルーチンを改版する際には、該サブルーチンの旧版はプログラム中に残したまま新版をプログラムに追加し、サブルーチン識別子とそのアドレスとの対応表において、該サブルーチンに対応するアドレスを、新版サブルーチンのものに書換えるだけでよい。したがって、システムを稼働停止すること無く容易にオンライン・バージョンアップを行うことができる。

【図面の簡単な説明】

【図 1】この発明の一実施例に係るコンピュータシステムの構成を示すブロック図。

【図 2】図 1 のコンピュータシステムに設けられたアドレス管理テーブルの具体的な構成の一例を示す図。

【図 3】図 1 のコンピュータシステムにおいて利用される関数呼び出し命令の一例を示す図。

【図 4】図 1 のコンピュータシステムにおいて関数 read file() を改版するための動作手順の一例を説明するフローチャート。

【図 5】図 1 のコンピュータシステムにおいて関数 read file() を改版するための他の動作手順を説明するフローチャート。

【図 6】図 1 のコンピュータシステムに設けられたアドレス管理テーブルの他の具体的な構成の一例を示す図。

【図 7】図 1 のコンピュータシステムにおいて図 2 のアドレス管理テーブルを用いて関数 read file() を呼出す手順を説明するフローチャート。

【図 8】図 7 のフローチャートにおけるテーブル参照動作を説明するフローチャート。

【符号の説明】

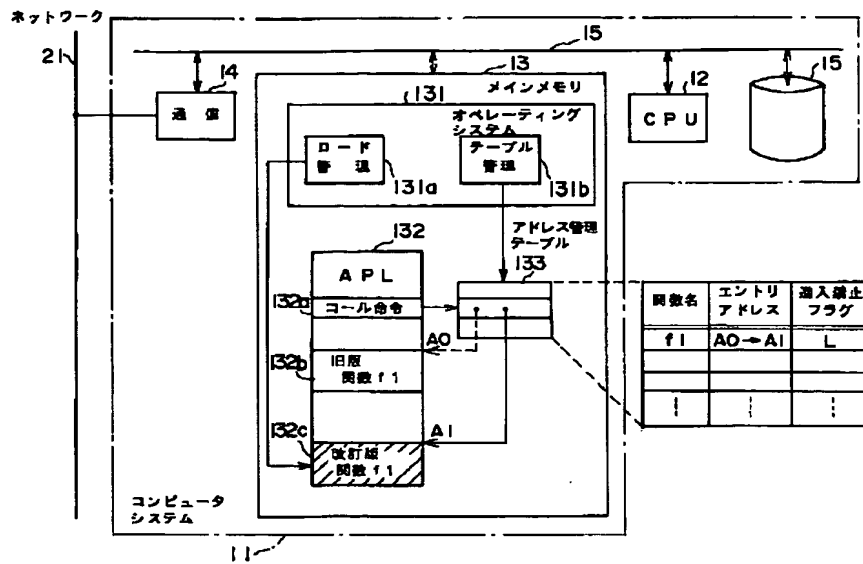
1 1…コンピュータシステム、1 2…CPU、1 3…メインメモリ、1 4…通信制御ユニット、1 5…磁気ディスク装置、1 3 1…オペレーティングシステム、1 3 1 a…サブルーチンロード管理プログラム、1 3 1 b…テーブル管理プログラム、1 3 2…アプリケーションプログラム、1 3 2 a…関数コール命令、1 3 2 b、1 3 2 c…関数、1 3 3…アドレス管理テーブル。

【図 6】

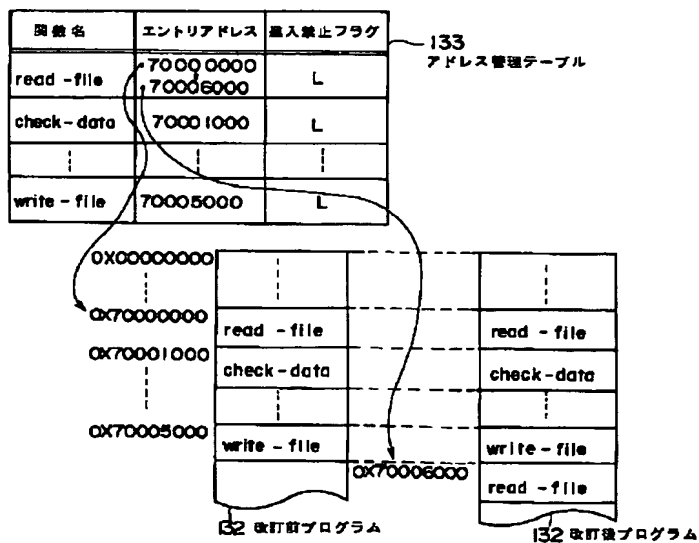
| 関 数 名 | エントリ アドレス | 進入禁止 フ ラ グ | スレッド カウンタ |
|-------|--------------|---------------|--------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

133 アドレス管理
テーブル

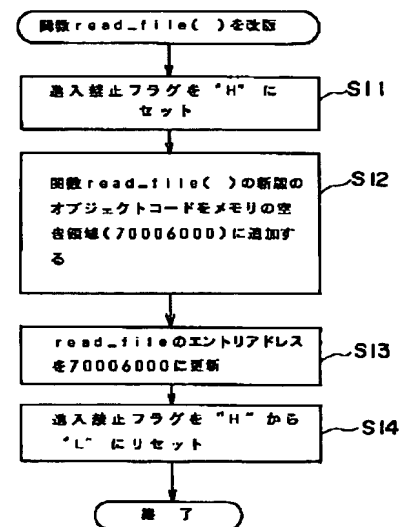
【図1】



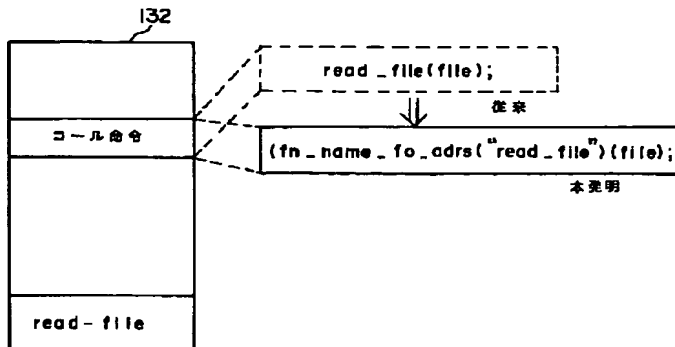
【図2】



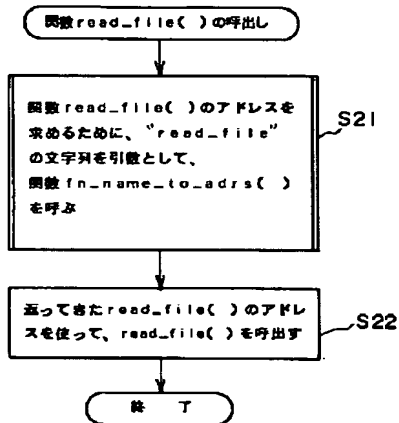
【図4】



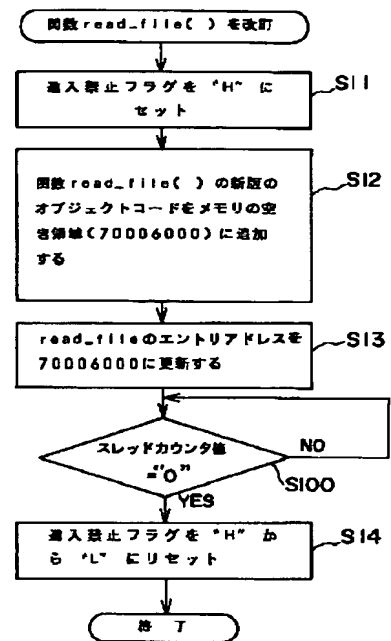
【図3】



【図7】



【図5】



【図8】

